ISO
TC184/SC4

WG11 N085

An Amendment to EXPRESS supporting STEP Modularization

An Update Since Lillehammer

September 1999

WG10 STEP Modularization PWI

Contact - David Price

dmprice@us.ibm.com

+1 843 760 4341

Agenda

- **Modularization and Amendment Requirements**
- **Technical Solutions**
- **The Standardization Plan**

Modularization and Amendment Requirements

Requirements

- **These requirements come from two sources**
  - **the WG10 STEP Modularization PWI**
  - **the nominated projects using the modular approach**
- **"Fundamental" high level requirements**
  - **These are the technical requirements on EXPRESS to support modularization itself**
- **"Amendment" high level requirements**
  - **These are the requirements on EXPRESS resulting from the fact that we are amending EXPRESS**

Requirements

- **Fundamental requirements**
  - upward compatibility
  - limit the impact on existing implementations
  - extensibility of constructed data types
  - separation of supertype constraint from entity
  - rename of attributes
- **Amendment requirements**
  - schema version identification
  - EXPRESS language version identification

Upward compatibility

- Any EXPRESS TC2 conforming schema shall remain valid under the amendment
- No changes identified to any existing SC4 schema shall be required as a result of the amendment
- Enhancements included in the amendment shall be compatible with EXPRESS edition 2
- These are requirements on the amendment itself
  - STEP Modularization is an "incremental improvement" therefore the upward compatibility of existing EXPRESS schemas is a requirement

Limit the impact on existing EXPRESS implementations

- **The scope of the amendment shall be as limited as is possible while meeting the modularization requirements**
- **No enhancement shall be included in the amendment that requires changes to Part 21 syntax**
- **Limit the impact on EXPRESS parsers as much as possible**
  - **However, there will be impact on parsers**

Extensibility of constructed data types

- **Modularization requires the ability to declare constructed data types that may be extended**
- **Modularization requires the ability to declare an extensible constructed data type that declares no items**
  - **e.g. empty extensible select type**

Separation of supertype constraint from entity

- **Modularization requires the ability to declare a supertype constraint outside the declaration of an entity**

Rename of attribute

- **Modularization requires the ability to rename an entity attribute as well as the existing ability to rename entities**

Schema version identification

- **Amending EXPRESS requires the ability to identify the version of a schema**
  - **Note that this also supports a requirement to configuration manage the numerous schemas that may result from modularization**

EXPRESS language version identification

- **Amending EXPRESS requires the ability to identify the version of the EXPRESS language to which the schema conforms**

Technical Solutions

Technical Solution - Use EXPRESS 2

- **Adopt EXPRESS edition 2 solutions with limitations**
  - **the amendment shall include only E2 solutions for which modularization has identified immediate requirements**
  - **the amendment shall not include E2 solutions that cause existing schemas to become invalid**
    - **The big exception is that we are adding new keywords so some schemas may in fact be invalid if they use those words and EXPRESS identifiers for entity, type, rule, etc.**
- **The syntax provided in this presentation is drawn from a working draft of the amendment and therefore provisional**

Technical Solution - Constructed Types

- **Extensibility of constructed data types**
  - **SELECT**

```
284 select_type = [ EXTENSIBLE ] [ GENERIC_ENTITY ]
  SELECT [  ( select_list | select_extension ) ] .
 410 select_list = '(' named_type { ',' named_type }
  ')' .
 409 select_extension = BASED_ON type_ref [ WITH
  select_list ] .
```

  - **ENUMERATION**

```
201 enumeration_type = [ EXTENSIBLE ] ENUMERATION [ (
  OF enumeration_items ) | enumeration_extension ] .
402 enumeration_items = '(' enumeration_id { ','
  enumeration_id } ')' .
403 enumeration_extension = BASED_ON type_ref [ WITH
  enumeration_list ] .
```

Example - Enumeration Types

```
SCHEMA s1;
TYPE general_approval = EXTENSIBLE ENUMERATION OF (approved,
   rejected);

SCHEMA s2;
USE FROM s1 (general_approval);
TYPE domain2_approval = EXTENSIBLE ENUMERATION BASED_ON
   general_approval WITH (rejected, pending);

SCHEMA s3;
USE FROM S1 (general_approval);
TYPE domain3_approval = EXTENSIBLE ENUMERATION BASED_ON
   general_approval WITH (cancelled);

SCHEMA s4;
USE FROM s2 (domain2_approval);
REFERENCE FROM S3 (domain3_approval);
TYPE specific_approval = ENUMERATION BASED_ON domain2_approval
   WITH (rework);
```

Example - Enumeration Types

In the context of schema S1:
general_approval has the domain **(approved, rejected).**

In the context of schema S2:
rejected in S2 is considered the same enum value as **rejected** in S1;
general_approval has the domain **(approved, rejected, pending);**
domain2_approval has the domain **(approved, rejected, pending).**

In the context of schema S3:
general_approval has the domain **(approved, rejected, cancelled);**
domain3_approval has the domain **(approved, rejected, cancelled).**

In the context of schema S4:
general_approval has the domain **(approved, rejected, pending, cancelled, rework);**
domain2_approval has the domain **(approved, rejected, pending, rework);**
domain3_approval has the domain **(approved, rejected, cancelled);**
specific_approval has the domain **(approved, rejected, pending, rework).**

Technical Solution - Supertype constraint

- Separation of supertype constraint from entity
  - Continue to allow existing SUPERTYPE constraint
  - Add E2 SUBYPE_CONSTRAINT which includes
    - The existing SUPERTYPE constraint constructs
    - TOTAL_OVER
    - ABSTRACT ENTITY
      - And the associated GENERIC data types
  - Note that E2 connotational subtype is not included as there is no requirement from modularization

Technical Solution - The new subtype syntax

```
412 subtype_constraint_decl = subtype_constraint_head
   subtype_constraint_body END_SUBTYPE_CONSTRAINT ';' .
413 subtype_constraint_head = SUBTYPE_CONSTRAINT
   subtype_constraint_id FOR entity_ref ';' .
411 subtype_constraint_body = [ abstract_supertype ] [
   total_over ] [ supertype_expression ';' ] .

400 abstract_supertype = ABSTRACT SUPERTYPE ';' .
415 total_over = TOTAL_OVER '(' entity_ref { ',' entity_ref }
   ')' ';' .
298 supertype_expression = subtype_factor { ANDOR
   subtype_factor } .
299 supertype_factor = supertype_term { AND supertype_term } .
301 supertype_term = entity_ref | one_of | '('
   supertype_expression ')' .
250 one_of = ONEOF '(' supertype_expression { ','
   supertype_expression } ')' .
```

Example - The new subtype syntax

```
ENTITY person;
ENTITY male SUBTYPE OF (person);
ENTITY female SUBTYPE OF (person);
ENTITY step_expert SUBTYPE OF (person);
ENTITY xml_expert SUBTYPE OF (person);

SUBTYPE_CONSTRAINT sub_mandatory_and_more
  FOR person;
  ABSTRACT SUPERTYPE OF
  ONEOF(male,female)
  TOTAL_OVER( male, female);
END_SUBTYPE_CONSTRAINT;
```

Example - The new subtype syntax

So:

A person cannot exist without also being one of its subtypes.

A person cannot be both a male and a female.

Male and female are “total coverage” of a concept and all subtypes of person must be one or the other.

Some person are step experts, some person are xml experts and some person may be both and some person may be neither.

Results:

male, female, male-step, male-xml, male-step-xml, female-step, female-xml, female-step-xml

Example - Abstract entity, Generic and Select Types

```
TYPE approvable_objects = EXTENSIBLE GENERIC_ENTITY SELECT;
END_TYPE;

ENTITY approval ABSTRACT;
  approved_by : GENERIC_ENTITY;
  status : approval_status_values;
  approved_items : SET[1:?] OF approvable_objects;
END_ENTITY;
```

IR

```
TYPE my_approvable_objects = SELECT BASED_ON
   approvable_objects WITH ( product, product_category,
   product_to_category_relationship );
END_TYPE;

ENTITY approval_by_person_in_organization SUBTYPE OF (
   approval );
  SELF\approval.approved_by : person_in_organization;
END_ENTITY;
```

AP

Technical Solution - Rename an attribute

- **Use ability to rename redeclared attributes**

```
167 attribute_decl = attribute_id |
redeclared_attribute
406 redeclared_attribute = qualified_attribute
[ RENAMED attribute_id ] .
354 qualified_attribute = SELF group_qualifier
attribute_qualifier .
295 group_qualifier = '\' entity_ref .
218 attribute_qualifier = '.' attribute_ref .
```

Example - Rename an attribute

```
ENTITY binary_entity_relationship ABSTRACT;
  end_one : GENERIC_ENTITY;
  end_two : GENERIC_ENTITY;
END_ENTITY;
ENTITY product_to_category_relationship SUBTYPE OF (
   binary_entity_relationship );
  SELF\binary_entity_relationship.end_one RENAMED the_category
   : product_category;
  SELF\binary_entity_relationship.end_two RENAMED the_product
   : product;
END_ENTITY;
ENTITY person_in_organization_relationship SUBTYPE OF (
   binary_entity_relationship );
  role_of_person : STRING;
  SELF\binary_entity_relationship.end_one RENAMED the_person :
   person;
  SELF\binary_entity_relationship.end_two RENAMED
   the_organization : organization;
END_ENTITY;
```

Technical Solution - Schema version identification

- Schema version identification

```
281 schema_decl = SCHEMA schema_id [
  schema_version_id ] ';' schema_body END_SCHEMA ';'
  .
407 schema_version_id = string_literal .
313 use_clause = USE FROM schema_ref [
  schema_version_ref ] [ '(' named_type_or_rename {
  ',' named_type_or_rename } ')' ] ';' .
408 schema_version_ref = string_literal .
267 reference_clause = REFERENCE FROM schema_ref [
  schema_version_ref ] [ '(' resource_or_rename {
  ',' resource_or_rename } ')' ] ';' .
```

Example - Schema version identification

```
SCHEMA geometry_schema version_1;
END_SCHEMA;

SCHEMA geometry_schema version_2;
END_SCHEMA;

SCHEMA config_controlled_design {ISO standard 10303
   part(203) version(3) object(2)};
  USE FROM geometry_schema version_2;
END_SCHEMA;
```

Technical Solution - Language version identification

- **EXPRESS language version identification**

```
405 language_version_id = '{ iso standard
  10303 part (11) version (4) }' .
302 syntax = [ language_version_id
  ]schema_decl { schema_decl } .
```

Example - Language version identification

```
{ iso standard 10303 part (11) version (4) }
SCHEMA config_controlled_design
{ISO standard 10303 part(203) version(3) object(2)};
  USE FROM geometry_schema version_2;

ENTITY approval ABSTRACT;
END_ENTITY;
...
END_SCHEMA;
```

ISO
TC184/SC4

WG11 N085

The Standardization Plan

Project management

- **Coordinate this project under EXPRESS 2**
- **Propose a New Work Item as soon as possible**
  - **expect placement in WG11**
  - **Project leader is same as the EXPRESS edition 2 project leader - Phil Spiby**
  - **Editors representing modularization and EXPRESS expertise - David Price and John Valois**
  - **Amendment is a stepping stone towards EXPRESS edition 2 supporting STEP modularization requirements - it DOES NOT remove need for E2!**

Standardization(1)

- Precede the EXPRESS edition 2 ballot
- According to ISO rules
  - no NWI, the amendment falls under the E2 NWI
  - cannot amend an IS with a TS, options are:
    - may be standardized as a “Minor Technical Amendment” which can go to directly into FDIS ballot with WG/SC4 approval
    - standardized as a “Minor Technical Revision” which is an entire new document is also possible
    - standardize as TS under an ISO number other than ISO 10303
    - normal CD-DIS-FDIS-IS ballot cycle
    - others?

Standardization(2)

- The project team recommends
  - 1999-08 Circulate presentation/summary information to SC4
  - 1999-09 Circulate the complete amendment with New Orleans SC4 resolution packet
  - 1999-11 "Ballot" workshop at New Orleans ISO
  - 1999-11 Proposed resolution for New Orleans SC4 meeting allowing "Minor Technical Amendment" going to FDIS ballot
    - This assumes all issues successfully addressed at New Orleans "ballot" resolution workshop, SC4 approval and appropriate QC review after New Orleans
  - 2000-02 If FDIS fails, repeat the process

Standardization(3)

- **Originally, the team recommended:**
  - **Assuming approval, publish the Technical Specification as a single document subsuming TC1, TC2 and the Amendment**
  - **Based on the Minor Technical Amendment process, combining the TCs and Amendment into one document is not possible**